Expense tracker

```
// App.js
import React, { useState } from "react";
import {
  SafeAreaView,
  View,
  Text,
  TextInput,
  Button,
  ScrollView,
  StyleSheet,
} from "react-native";

export default function App() {
  const [amount, setAmount] = useState("");
  const [category, setCategory] = useState("Food");
  const [day, setDay] = useState("Mon");
  const [records, setRecords] = useState([]);

  const categories = ["Food", "Clothes", "Bills", "Others"];
  const days = ["Mon", "Tue", "Wed", "Thu", "Fri", "Sat", "Sun"];

  const addRecord = () => {
    if (!amount || isNaN(amount)) return;
    const newRecord = {
      id: Date.now().toString(),
      category,
      amount: parseFloat(amount),
      day,
    };
    setRecords([...records, newRecord]);
    setAmount("");
  };
```

```
// Category-wise totals for weekly summary
const getSummary = () => {
  let summary = {};
  records.forEach((rec) => {
    if (!summary[rec.category]) summary[rec.category] = 0;
    summary[rec.category] += rec.amount;
  });
  return summary;
};

// Total weekly expense
const getTotal = () => {
  return records.reduce((sum, rec) => sum + rec.amount, 0);
};

return (
  <SafeAreaView style={styles.container}>
    <ScrollView contentContainerStyle={styles.scrollContent}>
      <Text style={styles.heading}>Weekly Expense Tracker</Text>

      {/* Input form */}
      <View style={styles.form}>
        <TextInput
          style={styles.input}
          placeholder="Amount"
          keyboardType="numeric"
          value={amount}
          onChangeText={setAmount}
        />

        {/* Category selector */}
        <View style={styles.row}>
          {categories.map((cat) => (
            <Button
```

```
        key={cat}
        title={cat}
        onPress={() => setCategory(cat)}
        color={cat === category ? "green" : "gray"}
      />
    ))}
  </View>

  {/* Day selector */}
  <View style={styles.row}>
    {days.map((d) => (
      <Button
        key={d}
        title={d}
        onPress={() => setDay(d)}
        color={d === day ? "blue" : "gray"}
      />
    ))}
  </View>

  <Button title="Add Expense" onPress={addRecord} color="purple" />
</View>

{/* Records list */}
<Text style={styles.subtitle}>This Week’s Records</Text>
{records.length === 0 ? (
  <Text style={styles.noRecords}>No records yet</Text>
) : (
  records.map((rec) => (
    <Text key={rec.id} style={styles.record}>
      {rec.day} - {rec.category}: ₹{rec.amount}
    </Text>
  ))
)}
```

```
      {/* Weekly summary */}
      <View style={styles.summary}>
        <Text style={styles.subtitle}>Weekly Summary</Text>
        {Object.entries(getSummary()).length > 0 ? (
          Object.entries(getSummary()).map(([cat, total]) => (
            <Text key={cat} style={styles.summaryText}>
              {cat}: ₹{total}
            </Text>
          ))
        ) : (
          <Text style={styles.summaryText}>No category data</Text>
        )}
        <Text style={styles.total}>Total Weekly Expense: ₹{getTotal()}</Text>
      </View>
    </ScrollView>
  </SafeAreaView>
 );
}

const styles = StyleSheet.create({
 container: {
  flex: 1,
  backgroundColor: "#fff",
 },
 scrollContent: {
  padding: 20,
  paddingBottom: 40,
 },
 heading: {
  fontSize: 22,
  fontWeight: "bold",
  marginBottom: 15,
  textAlign: "center",
```

```
},
form: {
  marginBottom: 20,
},
input: {
  borderWidth: 1,
  borderColor: "#ccc",
  padding: 8,
  marginBottom: 10,
  borderRadius: 5,
},
row: {
  flexDirection: "row",
  flexWrap: "wrap",
  marginBottom: 10,
  justifyContent: "space-between",
},
subtitle: {
  fontSize: 18,
  fontWeight: "bold",
  marginTop: 15,
  marginBottom: 8,
},
noRecords: {
  fontSize: 16,
  color: "gray",
},
record: {
  fontSize: 16,
  marginBottom: 4,
},
summary: {
  marginTop: 20,
  padding: 10,
```

```
    borderWidth: 1,
    borderColor: "#ddd",
    borderRadius: 6,
    backgroundColor: "#f9f9f9",
  },
  summaryText: {
    fontSize: 16,
    marginBottom: 4,
  },
  total: {
    fontSize: 18,
    fontWeight: "bold",
    marginTop: 10,
    color: "purple",
  },
});
```